МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных Наук

Кафедра программирования и информационных технологий

Клиент – серверное приложение «Wish Box»

Курсовой проект

090304 Программная инженерия

Выполнили:

Агафонова Марина Владимировна, студентка 3 курса, 4 группы
Аленичев Александр Викторович, студент 3 курса, 4 группы
Семечев Данила Алексеевич, студент 3 курса, 4 группы

Проверил:
Тарасов Вячеслав Сергеевич, ассистент кафедры ПиИТ

Воронеж 2020

Отчёт по 2 аттестации.

В течение второй аттестации было сделано следующее:

Агафонова Марина:

- Back-end:
  - Настройка проекта и подключение swagger
  - Классы моделей
  - Поиск пользователя с возможностью применения фильтра
  - Добавление нового желания (добавление UserId и изображения)
  - Добавление аватара и его редактирование
  - Редактирование желания (замена изображения)
  - Список желаний пользователя
  - Главная страница (вывод записей о добавлении нового желания пользователей из списка подписок текущего пользователя)
  - Пометка “Подарено” у желания
- Front-end:
  - Layout для всех страниц
  - Страница для неавторизованного пользователя
  - Страница авторизации
  - Страница регистрации
  - Страница добавления нового желания
  - Страница редактирования желания
  - Страница с желаниями пользователя
  - Страница поиска пользователя
  - Главная страница
  - Страница редактирования профиля
  - Страница пользователя
- Диаграммы:
  - Use case
  - Диаграмма последовательности
  - Диаграмма состояния
- Загрузка приложения в Azure

Аленичев Александр:

- Back-end:
  - Редактирование профиля
  - Добавление желания (основная часть)
  - Редактирование желания (основная часть)
  - Удаление желания
  - Изменение пароля
  - Добавление комментария
  - Удаление комментария
  - Создание комментария в ответ другому комментарию
  - Список тех подарков, которые взял на себя текущий пользователь
  - Подключение аналитики к проекту
- Front-end:
  - Страница изменения пароля
  - Открытие формы с комментариями
  - Список тех подарков, которые взял на себя текущий пользователь
- Аналитика:
  - Воронка «Регистрация»

    - Воронка «Создание желания»
    - Воронка «Подписаться»
- Диаграммы:
    - Диаграмма развёртывания
    - Диаграмма (объектов)

Семечев Данила:

- Back-end:
    - Настройка маршрутизации
    - Система авторизации и регистрации
    - Модель регистрации и авторизации
    - Проверка на авторизацию
    - Рейтинг желаний пользователя
    - Класс модели WishRating + добавления в модели данных, необходимых для рейтинга
    - Страница профиля пользователей
    - Модель страницы профиля пользователей
    - Страница "мои подписки"
- Front-end:
    - Базовая вёрстка страницы профиля
    - Базовая вёрстка страницы «мои подписки»
- Диаграммы:
    - Диаграмма классов
    - Диаграмма взаимодействия
    - Диаграмма активности
- Аналитика:
    - Воронка «Авторизация»
    - Воронка: «Переход на страницу пользователя»
    - Воронка: «Оценка желаний»

Содержание

Введение

Все мы любим дарить и получать подарки. В этом есть что-то магическое. Но иногда бывает так, что полученный подарок вам не нравится, да и вы уже начинаете понимать, что подарили другому человеку не самую нужную или приятную для него вещь.

В таких случаях на помощь приходят всяческие сервисы подбора подарков, где другие люди уже составили какие-то подарочные наборы. Правда, обычно это бывает дорого и не всем по карману.

Также есть различные приложения по поиску подарков. Но хорошего приложения на данный момент не существует, а существующие не имеют нужного функционала

В данной курсовой работе рассматривается проблема создания веб-приложения, предоставляющего пользователям возможность публиковать список желаемых ими подарков, а также возможность дарить эти самые подарки другим людям.

1. Постановка задачи

Цель курсовой работы – реализовать клиент – серверное web приложение, позволяющий пользователям «делиться» своими пожеланиями в качестве подарков, а также позволяющий дарить желаемый предмет. Оно должно позволять пользователю:

- Создавать/редактировать/удалять своё “желание”.
- Отмечать подарок другого пользователя, который они хотят ему подарить, и удалять данную отметку.
- Подписываться через поиск по пользователям на другие профили и отписываться от получения новостей о добавлении новых желаний определённого пользователя.
- Оставлять комментарии к записи с желанием и удалять его.
- Изменять данные своего профиля, включая смену пароля.

2. Анализ предметной области

2.1 Глоссарий

- Желание (подарок) - запись, которую пользователь публикует на странице своего профиля в качестве отображения того, что он хотел бы получить в подарок.
- Подписчик - пользователь, чьи новые желания будут видны на главной странице текущего пользователя.

2.2 UML диаграммы

2.2.1 Диаграмма последовательности

Диаграмма последовательности для создания желания, рисунок 1

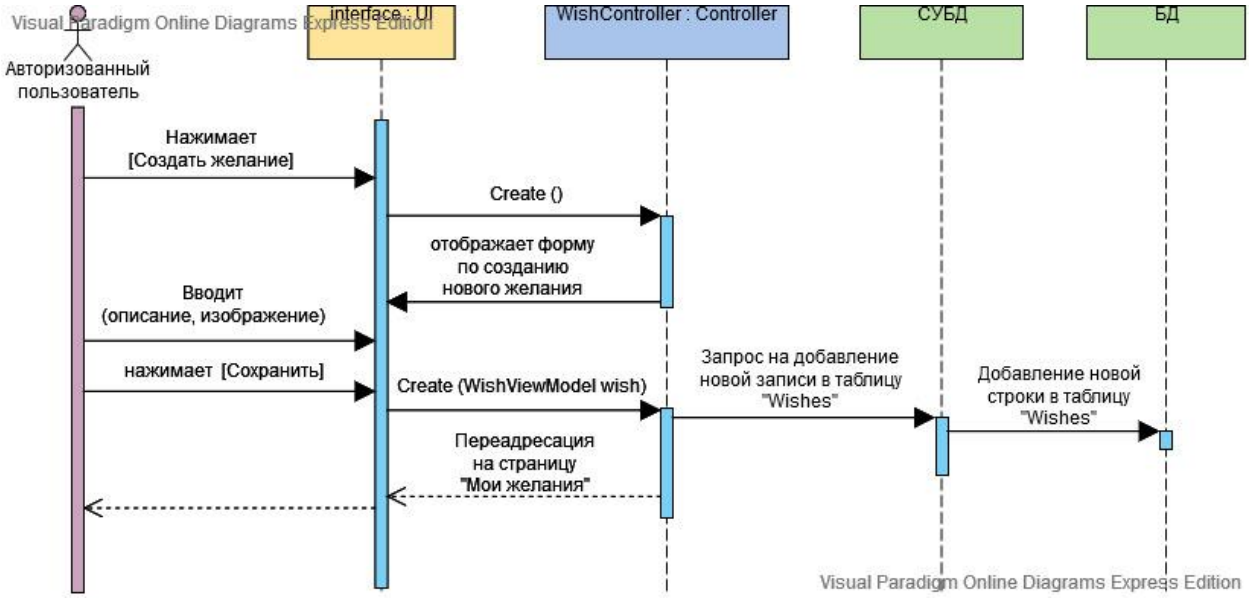

Рисунок 1. Диаграмма последовательности для создания желания

После взаимодействии пользователя, который обязательно должен быть авторизован, с интерфейсом приложения (нажатием соответствующей кнопки для создания желания) вызывается метод Create в контроллере, который возвращает форму по созданию нового желания. После введения описания желания и опциональной (необязательной) загрузки изображения для желания и нажатия кнопки «сохранить» вызывается метод Create (POST-запрос) с аргументом WishViewModel – моделью просмотра желания, которая содержит все данные, внесённые пользователем. Отравляется запрос в СУБД на добавление новой записи в таблицу «Wishes». В результате

добавляется новая строка в таблицу «Wishes» в базе данных. Происходит переадресация пользователя на страницу «Мои желания».

2.2.2 Диаграмма вариантов использования

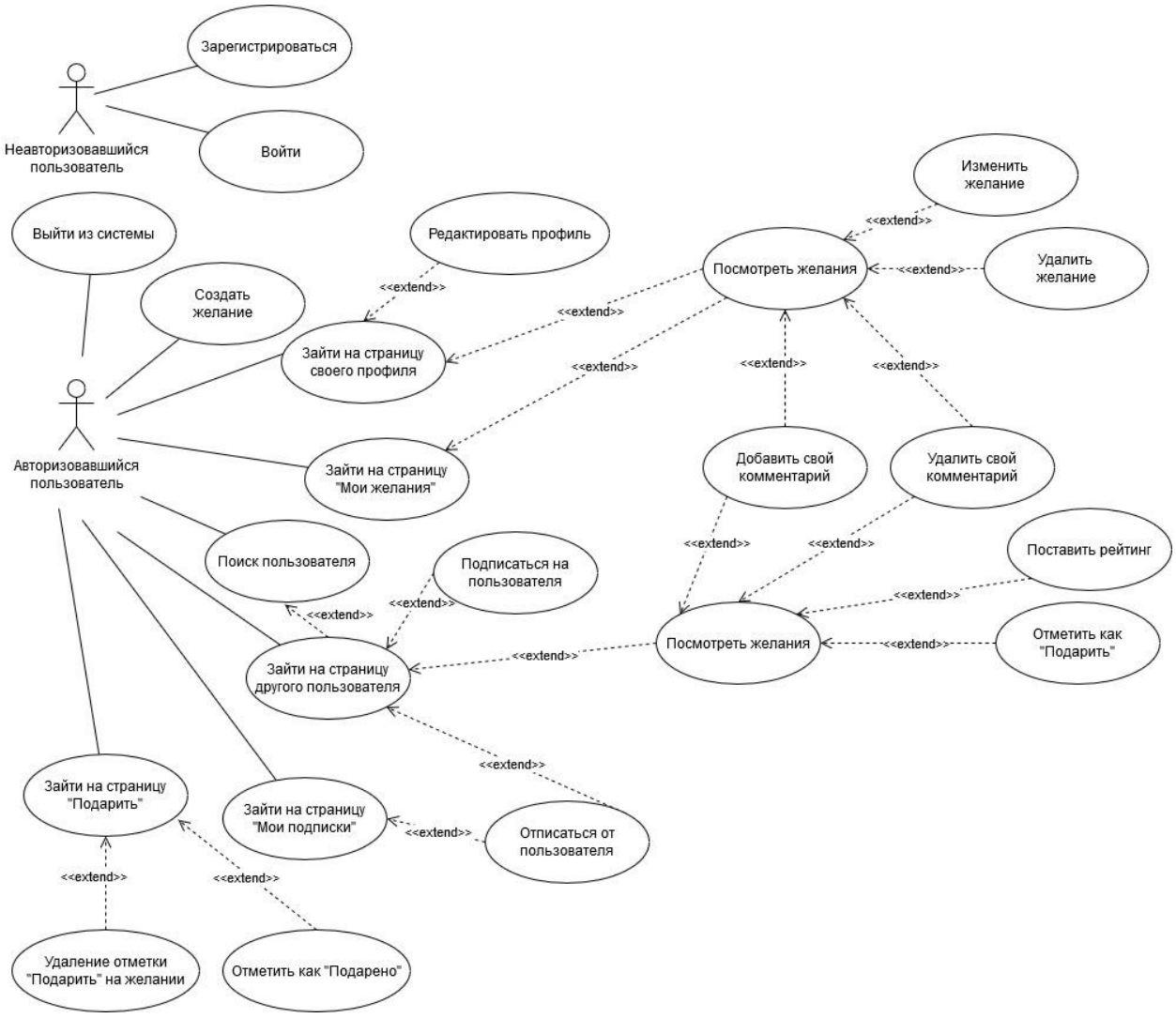

Рисунок 2. Диаграмма вариантов использования

Рассмотрим диаграмму вариантов использования для приложения на рисунке 2.

В системе могут быть два вида актора:

- Неавторизовавшийся пользователь
- Авторизовавшийся пользователь

Отношения ассоциации – действия, которые может осуществлять пользователь.

Для неавторизовавшего пользователя отношения ассоциации:

- Зарегистрироваться
- Войти

Для авторизовавшегося пользователя отношения ассоциации:

- Выйти из системы
- Создать желание
- Зайти на страницу своего профиля
- Зайти на страницу «Мои желания»
- Зайти на страницу другого пользователя
- Зайти на страницу «Мои подписки»
- Зайти на страницу «Подарить»

Для авторизовавшегося пользователя отношения расширения:

- Редактировать свой профиль
- Изменить своё желание
- Удалить своё желание
- Добавить комментарий
- Удалить свой комментарий
- Поставить рейтинг желанию другого пользователя
- Поставить отметку «Подарить» на желание другого пользователя
- Удалить отметку «Подарить» у ранее отмеченного желания.
- Поставить отметку «Подарено» на желание, которое есть на странице «Подарить»
- Подписаться на другого пользователя
- Отписаться от пользователя, на которого был ранее подписан текущий пользователь.

2.2.3 Диаграмма взаимодействия

Диаграмма взаимодействия для создания желания, рисунок 3

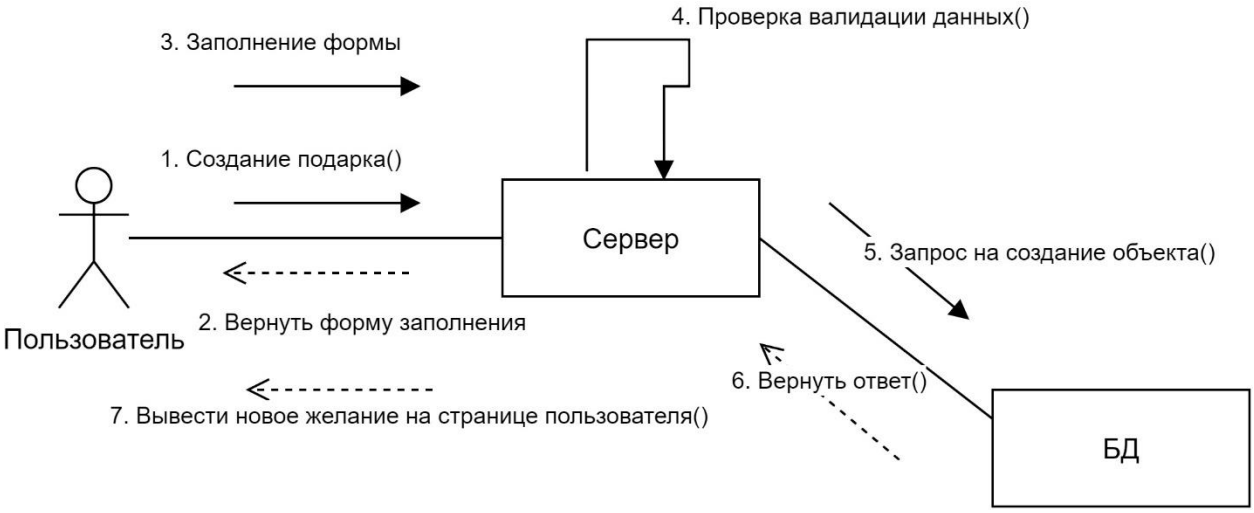

Рисунок 3. Диаграмма взаимодействия для создания желания

Для добавления нового желания пользователю необходимо на главном экране нажать кнопку «Добавить желание» (обозначена как + с подарком). После этого отображается окно для ввода данных о желании. Пользователю необходимо ввести данные и нажать кнопку «Сохранить». В случае успешно введенных данных, приложение делает запрос на сервер, который обращается в БД с запросом на сохранение. Когда данные сохранятся, сервер посылает ответ приложению. Затем пользователю отображается обновленный список желаний во вкладке “Мои желания”.

2.2.4 Диаграмма состояний

На рисунке 4 изображена диаграмма состояний.

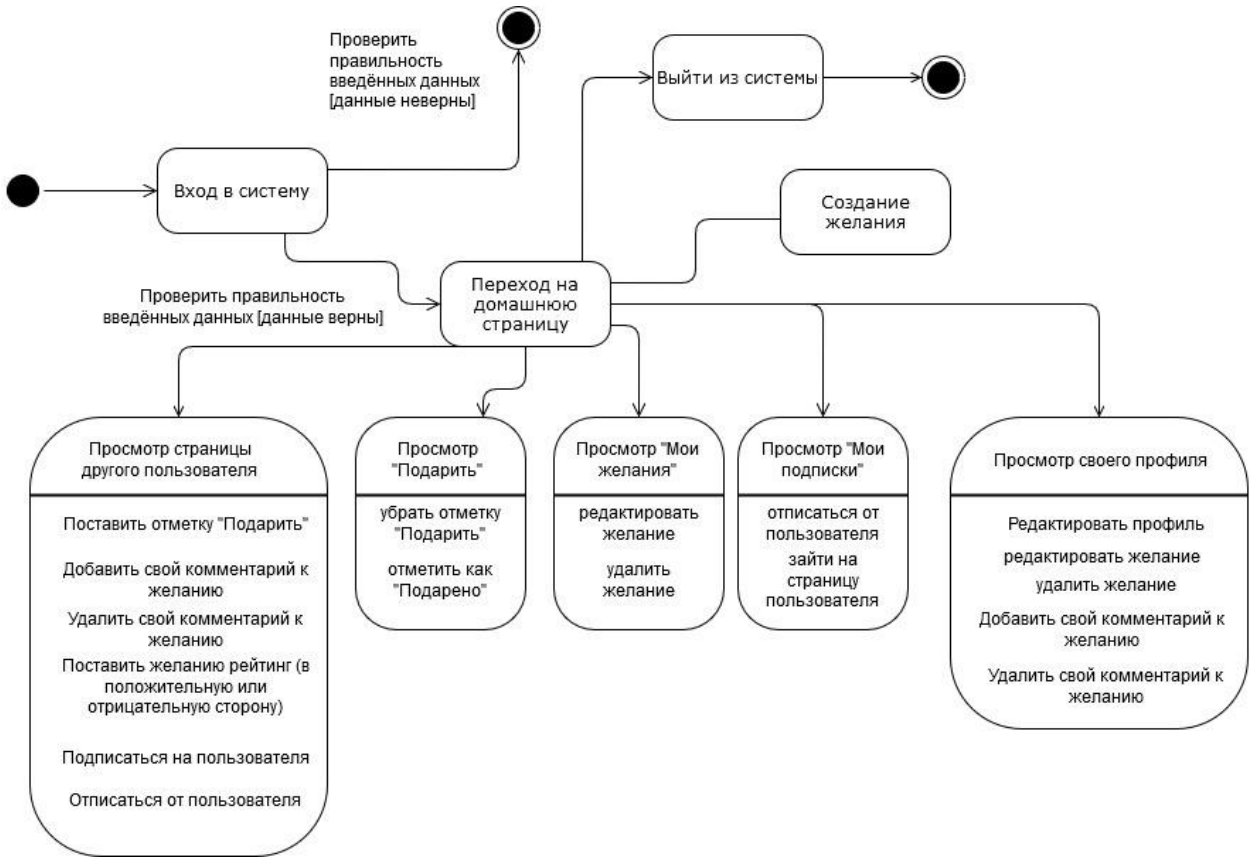

Рисунок 4. Диаграмма состояний

Диаграмма состояний отражает возможные состояния системы. При входе в систему осуществляется проверка корректности данных, которые ввёл пользователь для входа. В зависимости от результата проверки возможны 2 цепочки состояний, связанных с открытием доступа к функционалу приложения для пользователя.

Если результат проверки показал, что данные верны, система переадресовывает пользователя на домашнюю страницу, где для него доступен необходимый функционал.

Если пользователь выбрал функцию показа страницы другого пользователя, система обрабатывает запрос, отображает пользователю запрошенную страницу, где текущему пользователю предоставляется соответствующий перечень функций.

Аналогично происходит и с другими цепочками состояний.

Из любого состояния системы, кроме входа в систему, пользователь может выйти из системы.

2.2.5 Диаграмма развёртывания

На рисунке 5 представлена диаграмма развёртывания, чтобы определить, какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и какие различные части этого комплекса соединяются друг с другом.

Для развёртывания приложения требуется сервер с высокоскоростным доступом в интернет.Пользователю для взаимодействия с приложением требуется персональный компьютер с доступом в интернет, а также устройствами ввода-вывода информации, такие как клавиатура и мышь или сенсорная панель и монитор (экран).

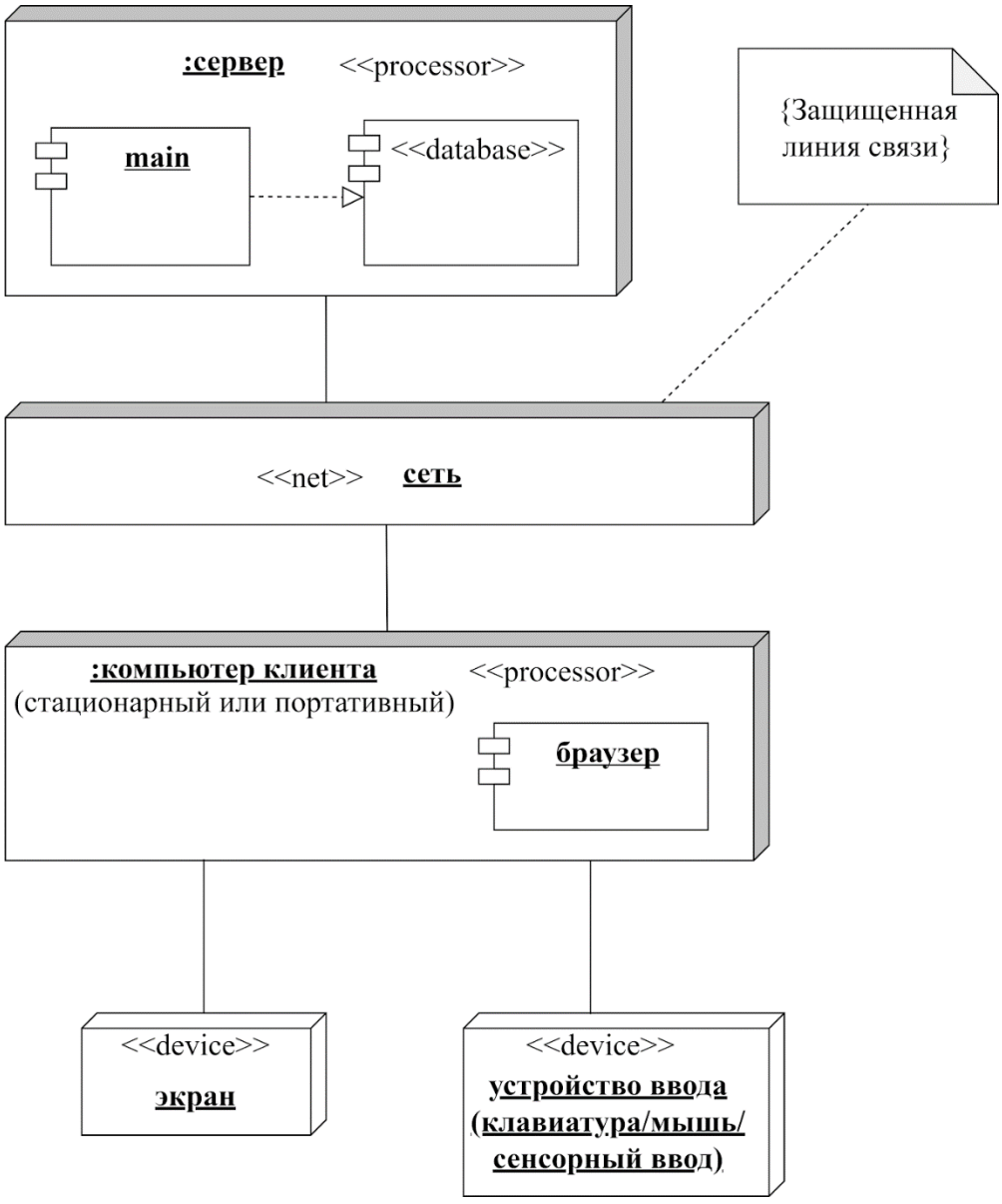

Рисунок 5. Диаграмма развёртывания

2.2.6 Диаграмма объектов

На рисунке 6 представлена диаграмма объектов

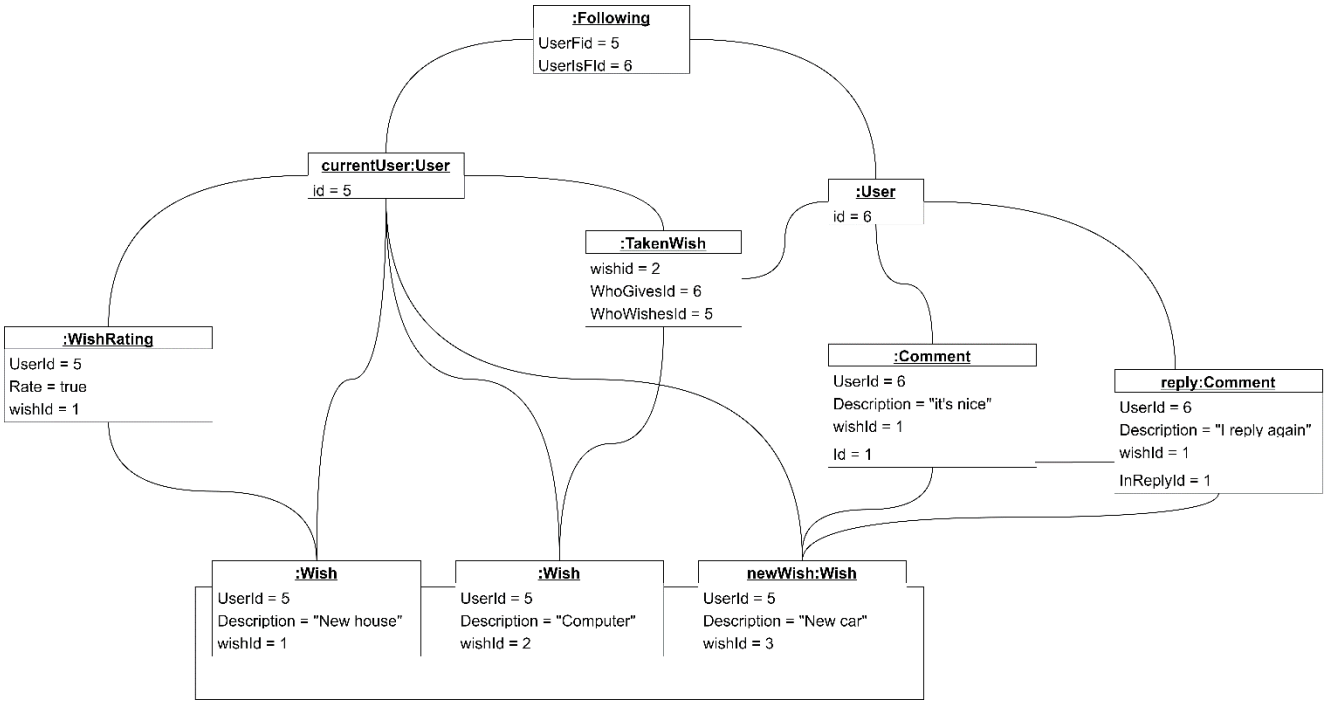

Рисунок 6. Диаграмма объектов

Пусть в некоторый момент времени существуют следующие объекты:

- экземпляр класса User с именем currentUser выступающий в роли текущего авторизованного пользователя
- 3 экземпляра “желаний” currentUser’а
- анонимный экземпляр класса User
- экземпляр класса WishRating, хранящий информацию о том, что текущий пользователь нажал на кнопку “+” рейтинга своего желания
- экземпляр класса Following, хранящий информацию о том, что пользователь с Id = 6 подписался на новости currentUser
- экземпляр класса TakenWish, хранящий информацию о том, что пользователь с Id = 6 решил подарить одно из желаний currentUser и нажал соответствующую кнопку
- анонимный экземпляр класса Comment, хранящий информацию о комментарии, оставленным пользователем с Id = 6 под желанием пользователя currentUser

2.2.7 Диаграмма активности

На рисунке 7 изображена диаграмма активности

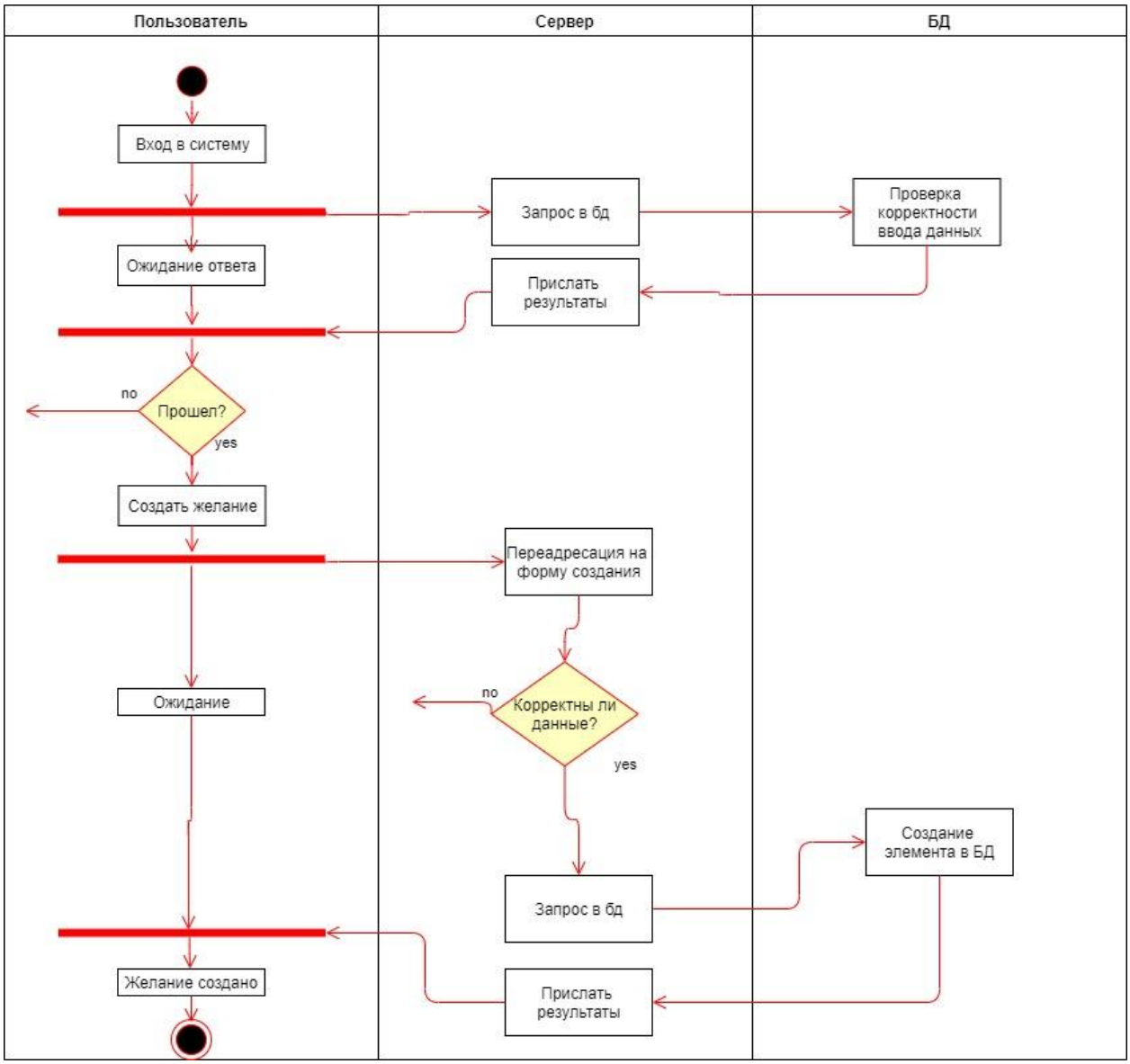

Рисунок 7. Диаграмма активности для создания желания

Авторизованный пользователь может добавить новое желание с главного экрана, нажав кнопку «Добавить желание» (обозначена как + с подарком). Приложение отобразит окно для ввода данных. Пользователю необходимо ввести данные о новом желании и нажать кнопку «Сохранить».

Далее приложение проверяет введенные данные, если данные корректны, приложение отправляет запрос на сервер. Если пользователь ввел неверные данные, приложение выведет пользователю ошибку. После чего пользователь может заново ввести данные или закончить действие. После отправки

запроса на сервер, сервер обрабатывает запрос и отправлять запрос к базе данных. После чего получает данные от БД и возвращает ответ. Приложение возвращает результат пользователю, и пользователь просматривает результат.

Если при обработке на сервере произошла ошибка или ответ от сервера не получен, приложение отображает пользователю ошибку, пользователь просматривает ошибку, после чего может попробовать еще раз или завершить действие.

2.2.8 Диаграмма классов

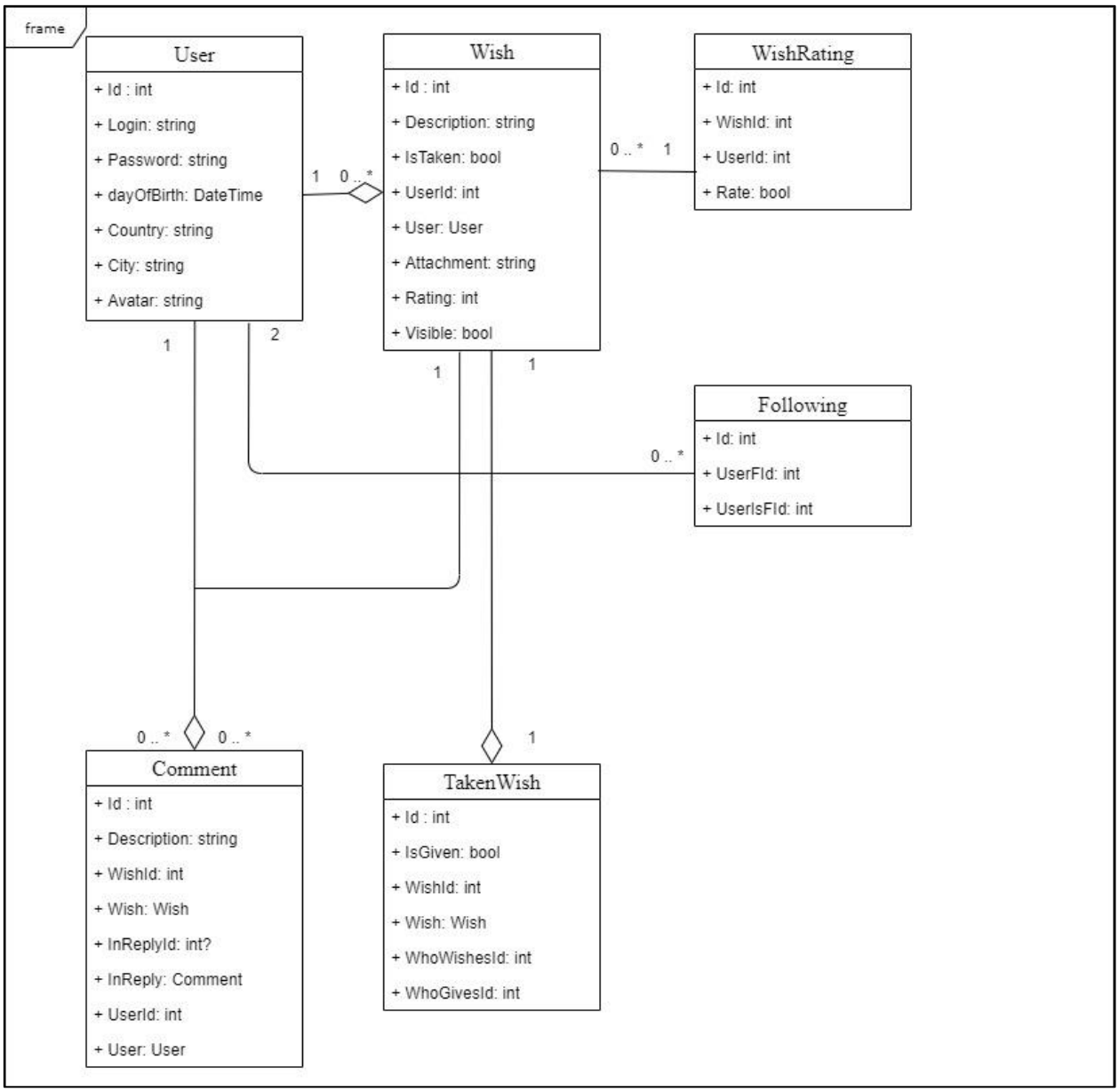

Рисунок 8. Диаграмма классов

Имеются несколько классов:

User, обозначающий пользователя приложения:

id - уникальный идентификатор

login - имя пользователя в системе

password - пароль

dayOfBirth - дата рождения

Country - страна проживания

City - город

Avatar - аватар пользователя;

Wish, обозначающий желания пользователей:

id - уникальный идентификатор

description - описание желания

IsTaken - хочет ли какой-то другой пользователь подарить этот предмет

UserId - уникальный идентификатор автора данного желания

User - автор желания

Attachment - изображение желания

Rating - рейтинг данного желания

IsVisible - видно ли это желание в приложении. Если у желания низкий рейтинг, то оно скрывается. По умолчанию все желания видны;

Following, обозначающий подписки пользователя на других пользователей:

id - уникальный идентификатор

UserFId - пользователь, который подписывается

UserIsFId - пользователь, на которого подписываются;

WishRating, обозначающий рейтинг(+ или -) того или иного желания от конкретного пользователя:

id - уникальный идентификатор

WishId - уникальный идентификатор желания

UserId - пользователь, который оценивает желание

Rate - сама оценка(“+” это true, “-” это false). Сделано для того, чтобы один пользователь не мог несколько раз поставить отрицательную или положительную оценку желанию;

WishRating был введён с целью контроля контента, который публикуют пользователь. Если рейтинг желания достигнет определённого порога, то оно будет удалено в связи с тем, что по мнению других пользователей представляет неуместный контент.

Comment, обозначающий комментарий под тем или иным желанием:

Id - уникальный идентификатор

Description - сам комментарий

WishId - уникальный идентификатор желания, под которым оставляется комментарий

Wish - само желание

InReply - дописать

InReplyId - дописать

UserId - уникальный идентификатор автора комментария

User - автор комментария;

TakenWish создан для сохранения информации о том, какой пользователь поставил отметку «Подарить» определённому желанию другого пользователя:

Id - уникальный идентификатор

IsGiven – флаг о том, что подарок был вручён автору желания

WishId - уникальный идентификатор желания, которое было отмечено как «подарить»

Wish - само желание

WhoWishesId - уникальный идентификатор автора желания

WhoGivesId – уникальный идентификатор пользователя, который хочет подарить данный подарок.

2.3 Схема базы данных

На рисунке 9 представлена ER-диаграмма, на которой показаны основные сущности системы и их взаимосвязь между собой.

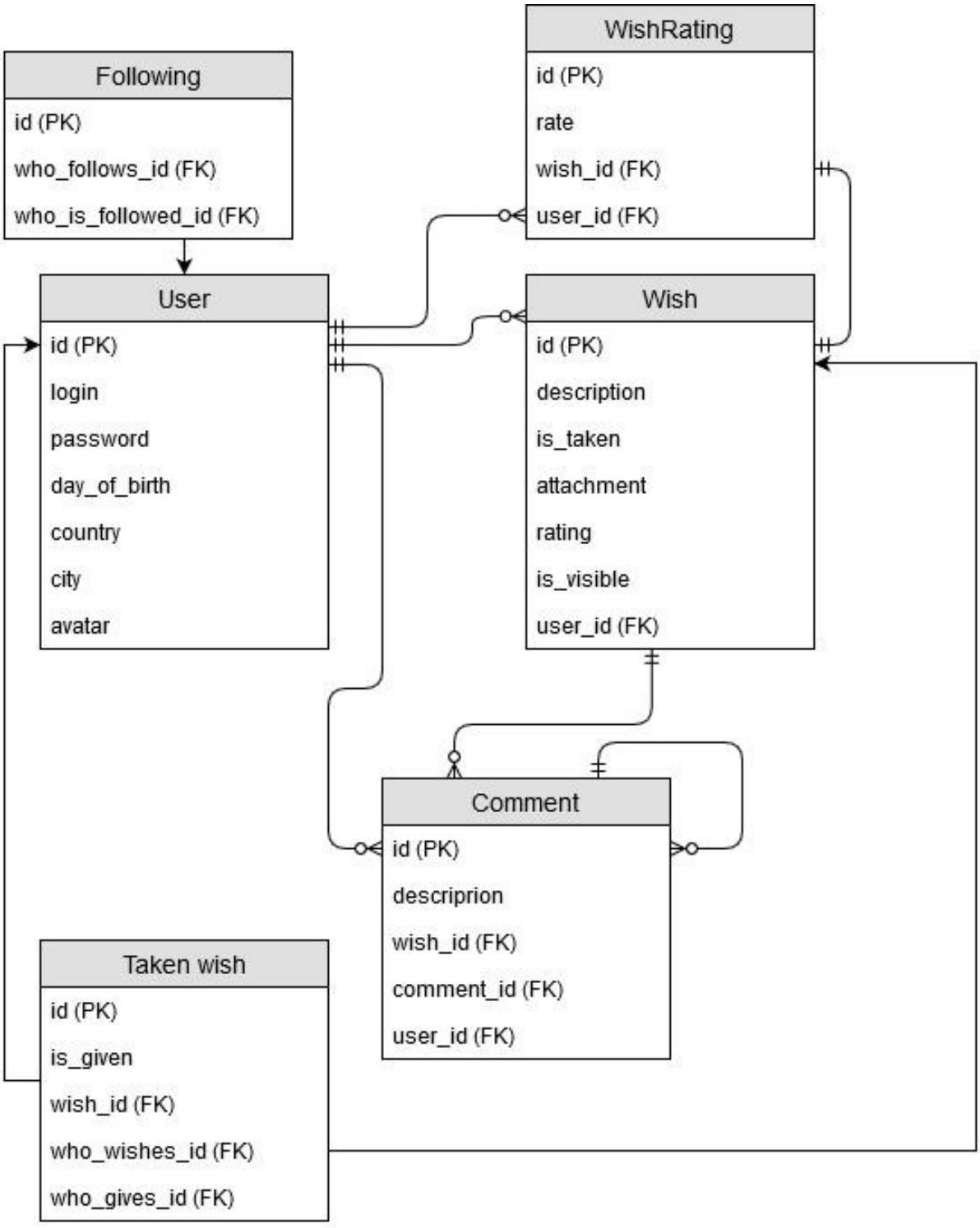

Рисунок 9. ER-диаграмма

- У пользователя может быть одно или несколько желаний (или не быть их совсем), а желания обязательно должен быть только один пользователь. Т.к. связь 1:m, в сущности “Wish” содержится внешний ключ “user_id”, который будет указывать на id пользователя, который создал данное желание.
- Т.к. к желанию можно написать комментарий, получается, что у желания может быть один или несколько комментариев (или может не быть вовсе), а комментарий должен обязательно ссылаться только на одно желание и только на одного пользователя. В функционале так же предусмотрен ответ на другой комментарий, поэтому будет существовать связь между сущностями “Comment” как 1:m, где у комментария может быть один или несколько ответных комментарий (или может не быть вовсе), а ответный комментарий обязательно должен ссылаться только на один комментарий.
- “Taken wish” – это соединяющая таблица для “User” и “Wish”, где хранятся данные о том, какое желание было отмечено, каким пользователем, кто создал данное желание и было ли оно подарено.
- “Following” – таблица, хранящая данные о том, какой пользователь на кого подписан.
- “WishRating” – таблица, хранящая данные о том, какой рейтинг (в большую сторону или в меньшую) поставил определённый пользователь определённому желанию другого пользователя. Конкретный рейтинг может относится только к одному желания, а пользователь может поставить рейтинг для многих желаний, а оценка пользователя относится только к этому конкретному пользователю.

3. Инструменты разработки

4. Описание работы программы

4.1 **Описание клиентской части приложения**

4.2 **Описание серверной части приложения**

5. Список использованных источников

6. Приложения